

# Kursportal – Struktur & HowTo

## Ordnerstruktur

```
senitkurs/  
  compose.yaml  
  web/  
    Dockerfile  
    requirements.txt  
    .env  
  app/  
    app.py  
    __init__.py  
    models.py  
    static/css/style.css  
    templates/  
      base.html  
      index.html  
      add_participant.html  
      edit_participant.html  
      list_participants.html  
      admin_home.html
```

## base.html & index.html

- base.html: Grundlayout (Head, Navigation, Footer), bindet static/css/style.css ein
- index.html: erbt von base.html, definiert nur title/content

Route:

```
@app.get("/")  
def index():  
    return render_template("index.html")
```

## Admin-Schutz (Token)

In .env:

```
ADMIN_TOKEN=dein_geheimes_token
```

Decorator in app.py:

```
def require_admin(f):  
    @wraps(f)  
    def decorated(*args, **kwargs):  
        token = request.args.get("admin") or request.headers.get("X-Admin-Token")  
        if token != ADMIN_TOKEN:  
            abort(403)  
        return f(*args, **kwargs)  
    return decorated
```

Admin-Hub:

```
@app.get("/_admin")  
@require_admin  
def admin_home():  
    return render_template("admin_home.html", token=ADMIN_TOKEN)
```

Aufruf: `http://localhost:5001/_admin?admin=DEIN_TOKEN`

## Wichtigste URLs

Öffentlich:

```
/
/teilnehmende/count
```

Admin (Token benötigt):

```
/_admin?admin=TOKEN
/teilnehmende?admin=TOKEN
/teilnehmende/new?admin=TOKEN
/teilnehmende/<id>/edit?admin=TOKEN
```

## Compose Kurz

```
docker compose up -d
docker compose build webapp && docker compose up -d webapp
docker compose logs -f webapp
```

## Nächste Schritte

- Zahlungen (Modell + Formular + Liste)
- Kurse/Blöcke/Termine als eigene Modelle/Seiten
- Optional: Login pro Teilnehmer:in (später)